

(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)	
Rajesh Venkat SUBBU et al.)	Group Art Unit 3693
)	Confirmation No. 5189
Serial No. 10/781,805)	
)	Examiner Jared W. Newton
Filed: February 20, 2004)	
)	Attorney Docket 141121-2
)	

For: SYSTEMS AND METHODS FOR MULTI-OBJECTIVE PORTFOLIO
ANALYSIS USING PARETO SORTING EVOLUTIONARY ALGORITHMS

APPEAL BRIEF

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

As required under § 41.37(a), this brief is filed within two months of the Notice of Appeal filed in this case on January 29, 2009, and is in furtherance of said Notice of Appeal.

Table of Contents

This Appeal Brief contains items under the following headings as required by 37 C.F.R.

§ 41.37 and M.P.E.P. § 1205.02:

I.	Real Party In Interest.....	3
II	Related Appeals and Interferences	4
III.	Status of Claims.....	5
IV.	Status of Amendments.....	6
V.	Summary of Claimed Subject Matter.....	7
VI.	Grounds of Rejection to be Reviewed on Appeal.....	17
VII.	Argument.....	18
	Rejection of Claims 27-36.....	18
	Rejection of Claims 27-54.....	20
VIII.	Claims Appendix.....	24
IX.	Evidence Appendix.....	30
X.	Related Proceedings Appendix.....	31

I. REAL PARTY IN INTEREST

The real party in interest for this Appeal is:

General Electric Company by way of an Assignment recorded at Reel/Frame
015679/0161 on August 12, 2004.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in this Appeal.

III. STATUS OF CLAIMS

A. Total Number of Claims in Application

There are 28 Claims pending in application.

B. Current Status of Claims

1. Claims canceled: 1-26.
2. Claims withdrawn from consideration but not canceled: None.
3. Claims pending: 27-54.
4. Claims allowed: None.
5. Claims rejected: 27-54.

C. Claims On Appeal

The Claims on Appeal are Claims 27-54.

IV. STATUS OF AMENDMENTS

In the Advisory Action dated January 27, 2009, the Examiner indicates that the Amendment After Final Rejection will be entered for purposes of Appeal. The status of the amendments to the Claims prior to filing the Notice of Appeal is as follows:

A. Responsive to a non-final Office action dated March 28, 2008, Appellant canceled Claims 1-26 and added Claims 27-54 on July 22, 2008.

B. Responsive to a final Office action dated October 29, 2008, Appellant amended Claims 47-54 on December 23, 2008.

C. Responsive to an Advisory Action dated January 27, 2009, Appellant timely filed a Notice of Appeal on January 29, 2009.

V. SUMMARY OF CLAIMED SUBJECT MATTER

In general, portfolio optimization conventionally involves trade-off between one return and one risk measure. In a quest for better risk management and investment decisions, modern portfolio managers have to simultaneously take multiple measures of often competing interests into account. These competing measures may typically be risk and return measures. From a portfolio optimization perspective in particular, one needs to solve a multi-objective optimization problem to obtain the efficient frontier of a particular investment problem. The objective functions, in particular risk measures, are far more complicated than those in a conventional approach, e.g., variance. Conventional analytical optimization approaches are not suitable for solving modern portfolio optimization problems, which involve trade-offs of complex objectives. A complex objective is defined as a function that is nonlinear and non-convex, or not in an analytical form. See *Paragraph [0086]*.

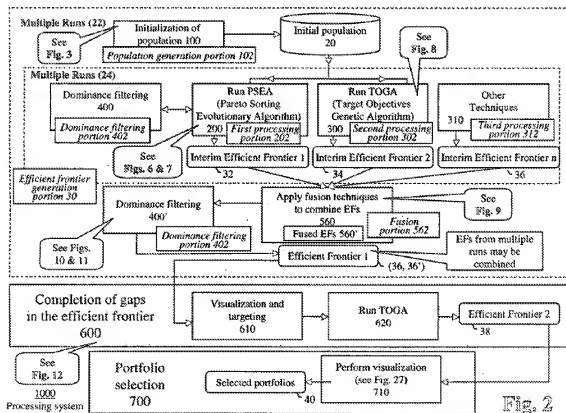
In the claimed invention, we propose a method for solving modern portfolio optimization problems with multiple objectives, some of which are complex functions. In accordance with one embodiment of the invention, the method uses evolutionary algorithms in conjunction with domain knowledge to determine the efficient frontier. The approach exploits domain knowledge via (1) the geometrical nature of the problem to explore and partially memorize the bounds of the feasible space and (2) the geometrical nature of the problem to design search operators that can efficiently explore the interiors of the feasible space, and (3) employs evolutionary algorithms for identifying the efficient frontier over multiple return and risk objectives, some of them being complex objectives. See *Paragraph [0087]*.

Figure 2 shown below provides an overall process chart of the invention, which may be performed by a suitable processing system 1000. As shown in Figure 2, the process starts in step 100 with an initialization of a population of solutions. This might be performed by a population generation portion 102, for example. The initial population is then output to an initial population database 20. After the initial population has been generated, the process of Figure 2 passes to what might be characterized as an efficient frontier generation portion 30, as shown in Figure 2. See *Paragraphs [0088] and [0089]*.

The efficient frontier generation portion 30 generates multiple interim efficient frontiers. Specifically, the efficient frontier generation portion 30 includes a number of processing portions (202, 302, 312), as desired, that perform efficient frontier generation processes. The processing portions (202, 302, 312) may perform the efficient frontier generation processes

using the same process or different processes. Further details of this processing are described below. As shown, the processing portions (202, 302, 312), as well as the other portions of Figure 2, may be disposed in the processing system 1000. See Paragraph [0090].

As a result of the processing of portions (202, 302, 312) multiple interim efficient frontiers (32, 34, 36) are generated. Figure 2 characterizes this processing as performing multiple runs 24. That is, each of the processing of portions (202, 302, 312) perform their own respective run so as to generate the respective efficient frontiers (32, 34, 36). As shown in Figure 2, these interim efficient frontiers are then output to a fusion portion 562. The fusion portion 562 performs a fusion process 560 on the multiple efficient frontiers (32, 34, 36), as described below, so as to generate a fused efficient frontier 560', i.e., an efficient frontier resulting from the fusing operation of step 560. See Paragraph [0091].



As shown in Figure 2, multiple runs 22 may be utilized to generate the efficient frontier 36. To explain further, it may be determined through examination of the efficient frontier 36, e.g., as a result of a first run, that the efficient frontier 36 is not sufficient. For example, it might

be, that the user senses that the generated efficient frontier 36 is not representative of the global situation. As a result, the process of Figure 2 might be run additional times, i.e., including generation of an initial population 20, generation of the interim efficient frontiers (32, 34, 36), fusing the interim efficient frontiers (32, 34, 36) and further dominance filtering 400' so as to generate a further efficient frontier 36'. Efficient frontiers (36, 36') from multiple runs may then be combined in some suitable manner, i.e., such as by simply adding the results together, if the user so desires. See *Paragraph [0095]*.

As described above, evolutionary algorithms, also known as genetic algorithms, are used to obtain solutions that are close to optimal by intelligently searching the feasible region. Evolutionary algorithms do this by implementing simplified models of natural evolution in a computer. In this approach populations of trial solutions compete for survival and succeed in proportion to their fitness, further participating in the production of offspring. The offspring with better genetic content (representation of a trial solution to the problem) survive and reproduce. After many generations of evolution, the best solutions are near optimal. See *Paragraph [0099]*.

The success of an evolutionary algorithm for an application depends in part on the computational efficiency of generating new solutions from parent solutions. The algorithm should be able to generate a new population of solutions from the previous generation in a reasonable amount of time. In the portfolio optimization problem, generating a valid portfolio is not trivial. This is because a typical portfolio has to satisfy a few thousand linear constraints. Ensuring that all the portfolios in a generation satisfy all these constraints requires computational run time. With known techniques, this makes using an evolutionary algorithm for solving the problem impractical. We therefore developed a novel method to circumvent this problem. See *Paragraph [00100]*.

In our evolutionary algorithm, we start with a large set of valid portfolios. This initial points population is generated using a novel technique, as described below in accordance with one embodiment of the invention, and is large enough to sufficiently sample the feasible region of portfolios. We generate each offspring, i.e., a valid portfolio in the next generation by using a convex combination of two or more parent portfolios (linear combination with non-negative weights that sum to unity). This is a convex crossover operation and allows quick generation of new populations from prior generations. To be able to search the entire feasible region using this convex crossover operation the initial points population should include all the vertices or

corner points of the polytope defined by the linear portfolio constraints. However, generating the corner vertices of a polytope is an NP-complete problem. That is, NP means “nondeterministic polynomial time”, and the term “nondeterministic” means that we need to guess what the solution might be. The solution can be tested in polynomial time, but the difficult aspect of such analysis is to guess right. NP-complete means “Nondeterministic Polynomial time-complete” and is a subset of NP, with the additional property that it is also NP-hard. Thus, a solution for one NP-complete problem would solve all problems in NP. Such a problem cannot be solved to completion in a reasonable time for any typical real-world portfolio. We have therefore developed a new algorithm to generate initial points that will sufficiently cover the search space for our needs. See *Paragraph [00101]*.

In accordance with one embodiment of the invention, the multi-objective portfolio optimization problem may be posed as:

$$\begin{aligned} & [\max f_1, \max f_2, \dots, \max f_n, \min g_1, \min g_2, \dots, \min g_m] \\ \text{subject to:} & \quad \text{Linear portfolio investment constraints,} \end{aligned}$$

where a function f_i is a return measure, and a function g_i is a risk measure. The goal is the joint maximization of return measures, and the minimization of risk measures. Each of these measures may be arbitrarily nonlinear and non-convex. However, the search space for this problem defined by the set of linear constraints is a bounded convex polytope. This polytope P is also the feasible search space. Any interior or extreme point of P is therefore a feasible portfolio investment strategy. See *Paragraph [00102]*.

Since a typical portfolio optimization problem can have thousands of constraints defined over a very high dimensional decision space, the feasible space P corresponding to valid portfolio investment strategies, though very large, will only have a small fraction of the volume of the smallest hypercube that encloses the feasible polytope, P . In this situation, starting out with a good idea of the extents of the feasible space P is a powerful feature. Such a feature also serves as a domain-specific guide to any evolutionary search over this space. In the absence of this information, an evolutionary search may consume significant time in either finding a feasible point, or in repairing an infeasible point to make it feasible. See *Paragraph [00103]*.

As shown in Figure 3 below, the processing that is used in the initialization of the population 100 starts in step 110. In step 110, the initial values of p_i and g_i are set such that the constraints involving them are always satisfied. For example, this can be done by setting all p_i s to zero and g_i s to large positive values. See *Paragraph [00107]*.

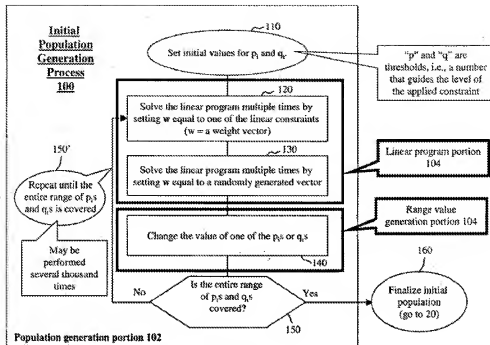


Fig. 3

Then, the process passes to step 120. In step 120, the linear program is solved multiple times by setting w equal to one of the linear constraints each time. Since in the resulting portfolio the linear constraint chosen for the objective function is activated, this process ensures that there is at least one portfolio where any specific constraint is activated. After step 120, the process passes to step 130. In step 130, the linear program is solved multiple times by setting w equal to a randomly generated vector. See Paragraph [00108].

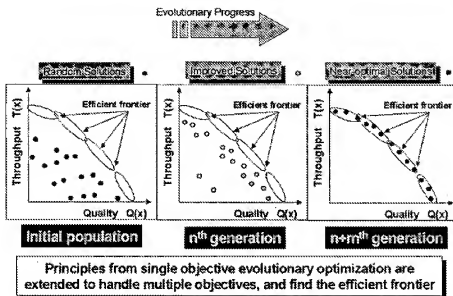
Then, the process proceeds to step 140, in accordance with this embodiment of the invention. In step 140, the value of one of the p_s or q_s is changed. Then, in step 150, a determination is made whether the entire range of p_s and q_s is fully covered, i.e., to the extent desired by the particular user. If no, then the process returns to step 120, and the processing is again performed as described above. Alternatively, if the entire range of p_s and q_s is fully covered, i.e., to the extent desired by the particular user, the process then passes to step 160. In step 160, the initialization processing is completed and the initial points population is output to the database 20, as shown in Figure 2. See Paragraph [00109].

As illustrated in Figure 3, the initialization of the population may be performed by a suitable processing system, i.e., by a population generation portion 102. The population

generation portion 102 may include a linear program portion 104 and a range value generation portion 106, in accordance with one embodiment of the invention. The linear program portion 104 performs the linear program processing as shown in steps 120 and 130 of Figure 3. Further, the range value generation portion 106 may change the values of the thresholds, as shown in step 140, as well as set the initial values for the competing objectives in step 110. As illustrated in Figure 3, the processing may be repeated several thousand times, for example, until a determination is made, by the population generation portion 102, that the entire range of competing objectives is covered. The particular threshold or criteria by which this determination is made may vary depending on the particular situation. The end result of the population generation portion 102 is the generation of an initial population of solutions for further processing, as described in detail below. See *Paragraphs [00110]-[00113]*.

The Pareto Sorting Evolutionary Algorithm (PSEA) is an approach to nonlinear non-convex multi-objective optimization, and provides an efficient method for portfolio risk optimization and planning, in accordance with one embodiment of the invention. The algorithm incorporates the basic evolutionary heuristics of selection, crossover (X-over), and diversity maintenance for identifying Pareto (efficient) frontiers in multi-objective optimization problems. See *Paragraph [00115]*.

The PSEA process of the invention works by systematically searching, memorizing, and improving populations of vectors (solutions), and performs multi-objective search via the evolution of populations of test solutions in an effort to attain the true Pareto frontier. A high-level view of the operation of the PSEA, a multi-objective optimization algorithm is shown in Fig 5. In this figure solutions are systematically evolved over multiple generations via structured operations in order to attain near-Pareto-optimal solutions. In this pictorial example, the goal is the identification of process plans that are Pareto optimal with respect to two objectives: quality and throughput. Starting from a population of random solutions, the PSEA is shown systematically closing in on those process plans that correspond to the true Pareto frontier as generations advance, i.e., effecting evolutionary progress. See *Paragraph [00116]*.

Fig. 5

The operational details of the “Run PSEA” step 200 of Figure 2 are hereinafter described through reference to the PSEA operational block diagram of Figure 6. That is, Figure 6 shows various populations and solutions utilized in the PSEA processing in accordance with one embodiment of the invention. Further, Figure 7 shows the process steps that are performed in conjunction with the processing of the populations and solutions of Figure 6. See *Paragraph [00122]*.

The processing of Figure 7 might be performed by an efficient frontier processing portion 202, in accordance with one embodiment of the invention. As shown in Figure 7, the process starts in step 200 and passes to step 1. In Step 1 of Figure 7, an initial population 210 of cardinality n is created by randomly drawn solutions. In the context of portfolio risk optimization, the solutions constituting the initial population are randomly drawn from a solutions archive 218 which may be comprised of the initial points population retrieved from a database, and which was generated using the initial points generation algorithm described above. Typically, the size of such a population would be in range of 50 to 100 solutions, for example. After step 1, the process passes to step 2. See *Paragraph [00123]*.

In Step 2 the initial population of n individuals is passed through a dominance filter 211. As a result of the dominance filtering, in step 3, a non-dominated subset of cardinality ($\mu < n$) is identified. In step 4 as shown in Figure 6, this non-dominated subset is committed to the non-

dominated solutions archive 220, as shown in Figure 6. After step 4, the process passes to step 5. See *Paragraph [00124]*.

In Step 5, $n/2$ randomly matched pairs of parent solutions from the population 210 are combined to create n “offspring” solutions. In the context of portfolio risk optimization, since the feasible space is convex, and we already include a sample of the bounds of the convex feasible space via points from the solutions archive, convex combination of two parent solutions to yield two offspring solutions is a suitable crossover technique. Given two vectors \mathbf{x} and \mathbf{y} , the two offspring vectors generated via convex crossover are $w\mathbf{x} + (1-w)\mathbf{y}$, and $(1-w)\mathbf{x} + w\mathbf{y}$, where w is a real random number in the space $[0, 1]$. The n offspring solutions are passed through a dominance filter in Step 6. The dominance filter processing identifies the non-dominated subset of cardinality $(\lambda < n)$, as shown in Step 7. See *Paragraph [00125]*.

Accordingly, as shown in Figure 6, the processing box 212 shows the prepared non-dominated parent solutions and the non-dominated off-spring solutions. In step 8, these two solutions are combined. That is, in step 8, we combine the two non-dominated solution subsets of cardinality $(\mu + \lambda < 2n)$. The combined solutions are passed through a non-crowding filter 214 in step 9. The non-crowding filter 214 removes a smaller subset of α solutions that are clustered. The result of this non-crowding operation of step 9 is a reduced solution subset of cardinality $\mu + \lambda - \alpha$. The non-crowding filter identifies regions that are heavily clustered and drops solutions from those dense clusters, i.e., so as to generate a more representative set of solutions. See *Paragraph [00126]*.

After step 9, the process passes to step 10, in accordance with one embodiment of the invention. In Step 10, a new population 216 of n individuals is created. If $n = \mu + \lambda - \alpha$, then the filtered population from Step 9 becomes the new improved population. In the case that $n > \mu + \lambda - \alpha$, m individuals from the solutions archive 218 are randomly drawn such that $n = m + \mu + \lambda - \alpha$. The inclusion of these m individuals from the solutions archive 218 injects new points into the population enhancing its diversity. See *Paragraph [00127]*.

Alternatively, it may be the case that $n < \mu + \lambda - \alpha$. In this situation, we are faced with a problem of discarding some solutions from the set of $\mu + \lambda - \alpha$, and at the same time promoting diversity via the injection of new points. This may be achieved in two sub-steps. In the first sub-step, $(n - p)$ solutions are randomly selected from the $\mu + \lambda - \alpha$ solutions, where “ p ” is a number that is one tenth the magnitude of n , for example. In this processing, the value of “ p ”

stays constant (e.g. “p” may be 5 solutions). In other words, β solutions (e.g. 20 solutions) are randomly discarded from the set of $\mu+\lambda-\alpha$ solutions (e.g., which might number 65). To adjust the cardinality of the new population to n, p solutions are randomly injected from the solutions archive. The inclusion of these p individuals from the solutions archive injects new points into the population enhancing its diversity, and resulting in a desired number of 50 solutions, for example. The process of Figure 6 and Figure 7 then passes to step 11. See *Paragraph [00128]*.

In Step 11 of Figure 7, the new population 216 replaces the previous population 210, i.e., the population 210 is updated, and the evolutionary process is continued, i.e., repeated, until convergence is achieved, or allocated computational cycles are exhausted. Accordingly, at the conclusion of the evolutionary search, a non-dominated solutions archive 220 is achieved. See *Paragraph [00129]*.

Accordingly, in step 12 of Figure 7, the archive of non-dominated solutions from each generation is finalized to result in a “Final-1” set of global non-dominated solutions. Then, in step 14, the archived non-dominated solutions from each generation are processed using dominance filtering 211, for example, as described in Figure 10 and Figure 11. After step 14, a “Final-2” set of filtered, global non-dominated solutions are generated. Then, in step 16 of Figure 7, the filtered, global non-dominated solution set is output to an archive 230. After step 16 of Figure 7, the process passes to step 18. In step 18, the process returns to Figure 2. That is, the processing of Figure 6 and Figure 7 results in the interim efficient frontier 32, which may be temporarily stored, as desired. In step 16, the process finalizes the resulting archive of filtered global non-dominated solutions. Accordingly, a near-Pareto-optimal frontier is generated. See *Paragraphs [00129] and [00130]*.

Fig. 6

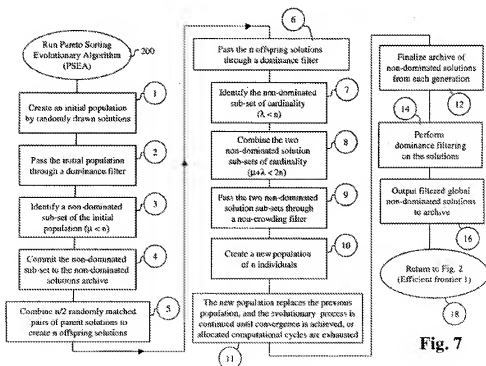
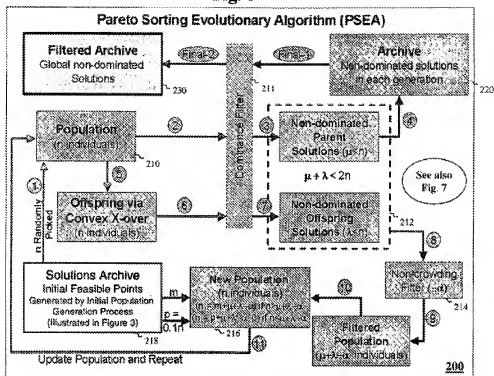


Fig. 7

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether Claims 27-36 are nonstatutory subject matter under 35 U.S.C. 101.
2. Whether Claims 27-54 are unpatentable under 35 U.S.C. §103(a) over Yao et al. (WIPO Publication No. 02/075650, hereinafter “Yao”) in view of Josephson et al. (U.S. Patent No. 7,155,423, hereinafter “Josephson”).

VII. ARGUMENT

1. Rejection of Claims 27-36 under 35 U.S.C. §101

Independent Claim 27 specifies, *inter alia*, a method for multi-objective portfolio analysis using Pareto Sorting Evolutionary Algorithms, the method comprising the steps of:

- (a) randomly drawing an initial population of individual portfolio allocations that are generated from a portfolio allocations archive by using a combination of linear programming and sequential linear programming algorithms using a computing device;
- (b) passing the initial population of portfolio allocations through a dominance filter to identify a non-dominated subset of parent portfolio allocations;
- (c) committing the non-dominated subset of parent portfolio allocations to a non-dominated portfolio allocations archive;
- (d) randomly combining matched pairs of parent portfolio allocations to create offspring portfolio allocations;
- (e) passing the offspring portfolio allocations through the dominance filter to identify a non-dominated subset of offspring portfolio allocations;
- (f) combining the non-dominated subset of parent portfolio allocations with the non-dominated subset of offspring portfolio allocations into a larger set of portfolio allocations;
- (g) passing the larger set of portfolio allocations through a non-crowding filter to identify a reduced subset of portfolio allocations;
- (h) creating a new population of individual portfolio allocations from the reduced subset of portfolio allocations;
- (i) updating the non-dominated portfolio allocations archive with the new population of individual portfolio allocations;
- (j) repeating steps (a) through (i) for a plurality of generations; and
- (k) passing the updated non-dominated portfolio allocations archive through the dominance filter to generate an interim efficient frontier in a portfolio performance space having at least three-dimensions, the interim efficient frontier being used in investment decisions. (Emphasis Added).

The relevant statute, 35 U.S.C. 101, provides:

“Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefore, subject to the conditions and requirements of this title.”

It is important to be reminded that 35 U.S.C. 101 was intended to render patentable “anything under the sun made by man,” assuming the other conditions of patentability are fulfilled. *Diamond v. Chakrabarty*, 447 U.S. at 309 (1980).

In *Bilski*, the Federal Circuit sitting en banc held that a process is patent-eligible if (1) it is tied to a particular machine or apparatus, or (2) it transforms a particular article into a different state or thing. *In re Bilski*, 545 F.3d 943, 88 U.S.P.Q.2d 1385 (2008). This is called the “machine-or-transformation test.” There are two corollaries to the machine-or-transformation test. First, a mere field-of-use limitation is generally insufficient to render an otherwise ineligible method claim patent-eligible. This means the machine or transformation must impose meaningful limits on the method claim’s scope to pass the test. Second, insignificant extra-solution activity will not transform an unpatentable principle into a patentable process. This means reciting a specific machine or a particular transformation of a specific article in an insignificant step, such as data gathering or outputting, is not sufficient to pass the test. See *Memorandum from John J. Love to the Technology Center Directors dated January 7, 2009*.

Appellant respectfully asserts that independent Claim 27 is tied to another statutory class and defines statutory subject matter. In particular, the first step of independent Claim 27 is directed to randomly drawing an initial population of individual portfolio allocations that are generated from a portfolio allocations archive by using a combination of linear programming and sequential linear programming algorithms using a computing device. (Emphasis added). When independent Claim 27 is viewed as a whole, Appellant asserts that it is impossible for the human mind to perform the complex plurality of steps recited in independent Claim 27, not besides the fact that the various complex steps are repeated for a plurality of generations in step (j). In other words, a computing device is necessary to perform the various complex steps of independent Claim 27, such as the processing system 1000. Thus, independent Claim 27 is tied to another statutory class and defines statutory subject matter under 35 U.S.C. 101, so the rejection should be reversed.

Even if, *arguendo*, independent Claim 27 is not considered to be tied to another statutory class, independent Claim 27 satisfies the second prong of the *Bilski* machine-or-transformation test by transforming the underlying subject matter to a different state or thing. Specifically, steps (c) and (i) of independent Claim 27 transforms the non-dominated portfolio allocations archive committed in step (c) to an updated non-dominated portfolio allocations archive in step (i). According to its plain meaning in the computer science art, an “archive” is defined as:

1. A long-term storage area, often on magnetic tape, for backup copies of files or for files that are no longer in active use.
2. A file containing one or more files in compressed format for more efficient storage and transfer. (www.dictionary.com).

Thus, the underlying subject matter, i.e., the non-dominated portfolio allocations archive, is being transformed or updated for each generation.

After updating the non-dominated portfolio allocations archive is repeated for a plurality of generations, the archive is passed through a dominance filter to generate an interim efficient frontier to be used in investment decisions. Thus, the updated non-dominated portfolio allocations archive can not be considered to be insignificant extra-solution activity.

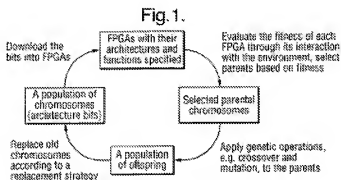
In view of the foregoing, Appellant respectfully submits that independent Claim 27 defines statutory subject matter, so the rejection of Claims 27-36 under 35 U.S.C. 101 should be reversed.

2. Rejection of Claims 27-54 under 35 U.S.C. 103(a) over Yao in view of Josephson

Independent Claims 27, 37 and 46 specify, *inter alia*, the feature of randomly drawing an initial population of individual portfolio allocations that are generated from a portfolio allocations archive by using a combination of linear programming and sequential linear programming algorithms using a computing device. (Emphasis added). Appellant respectfully submits that at least this feature is not disclosed, taught or suggested in the applied art, taken singly or in combination.

Yao discloses a hardware design process using evolution algorithms. As shown in Figure 1 below, an initial population of architecture bits encoded as chromosomes 10 are

generated either at random or heuristically. The chromosomes 10 are downloaded 12 into field programmable gate arrays (FPGAs) 14 for fitness evaluation. The fitness of an FPGA is evaluated through its interaction with the environment 16. Such fitness is then used to select parent chromosomes 18 for further reproduction and genetic operation. Crossover and mutation 20 are often used to generate offspring chromosomes 22 from the parents. These offspring will then replace their parents according to certain replacement strategy. Some replacement strategies may retain a parent and discard its offspring. A new generation of chromosomes is formed after replacement. See *Page 3, lines 7-23*.



On Page 3 of the final Office action, the Examiner asserts that:

“Yao discloses a method for multi-objective problem solving using evolutionary algorithms, said method comprising:

randomly drawing an initial population of individual solutions that are generated from an archive by using a combination of linear programming and sequential linear programming algorithms using a computing device (see claim 1);

passing the initial population of solutions through a dominance filter to identify a non-dominated subset of parent solutions (see claim 1; pages 16-22);

committing the non-dominated subset of parent solutions to a non-dominated solutions archive;

randomly combining matched pairs of parent solutions to create offspring solutions (see pages 16-22);

passing the offspring solutions through the dominance filter to identify a non-dominated subset of offspring solutions (see pages 16-22);

combining the non-dominated subset of parent solutions with the non-dominated subset of offspring solutions into a larger set of solutions (see pages 16-22);

passing the larger set of portfolio allocations through a non-crowding filter to identify a reduced subset of solutions in order to create a new population of individual solutions from the reduced subset of solutions, and updating the solution archive with the new population (see pages 32-36);

repeating the above steps for a plurality of generations (see claim 1); and

passing the updated non-dominated solutions archive through a dominance filter to generate an interim efficient frontier having at least three-dimensions, the frontier being used to make problem solution decisions (see claim 1).”

Claim 1 of Yao states the following:

1. A method of designing a hardware element using an evolutionary algorithm, comprising the steps of:
 - a) providing an initial population of hardware elements;
 - b) encoding the initial population as chromosomes;
 - c) evaluating the fitness of each of the initial population according to multi-objective fitness criteria;
 - d) selecting parent chromosomes based on the fitness evaluation of the initial population;
 - e) applying genetic operations to the selected parent chromosomes to produce a population of offspring;
 - f) selecting a set of new chromosomes from the parent and offspring chromosomes, comprising forming a plurality of clusters from the parent and offspring chromosomes and forming a Pareto front of non-dominated chromosomes for each cluster; and
 - g) repeating steps c) to f) for the new set of chromosomes to form a new generation until a predetermined termination criterion is satisfied.

Contrary to the Examiner, there is no mention whatsoever in Claim 1 of Yao of at least the feature of randomly drawing an initial population of individual solutions that are generated from an archive by using a combination of linear programming and sequential linear programming algorithms using a computing device, as recited in independent Claim 27 of the claimed invention. Josephson adds nothing to overcome this shortcoming in Yao.

For at least this reason, the Examiner fails to establish a *prima facie* case of obviousness, so the rejection of Claims 27, 37 and 46 is unsupported by the art and should be reversed. Claims 28-36, which depend from Claim 27, Claims 38-45, which depend from Claim 37, and Claims 47-54, which depend from Claim 46, are likewise allowable over the applied art, taken singly or in combination.

In view of the foregoing, Appellant respectfully submits that the application is in condition for allowance. Favorable consideration and prompt allowance of the application is earnestly solicited.

Dated: March 27, 2009

Respectfully submitted,

By /Peter J. Rashid/

Peter J. Rashid, Reg. No. 39464

VIII. CLAIMS APPENDIX

27. A method for multi-objective portfolio analysis using Pareto Sorting Evolutionary Algorithms, the method comprising the steps of:

- (a) randomly drawing an initial population of individual portfolio allocations that are generated from a portfolio allocations archive by using a combination of linear programming and sequential linear programming algorithms using a computing device;
- (b) passing the initial population of portfolio allocations through a dominance filter to identify a non-dominated subset of parent portfolio allocations;
- (c) committing the non-dominated subset of parent portfolio allocations to a non-dominated portfolio allocations archive;
- (d) randomly combining matched pairs of parent portfolio allocations to create offspring portfolio allocations;
- (e) passing the offspring portfolio allocations through the dominance filter to identify a non-dominated subset of offspring portfolio allocations;
- (f) combining the non-dominated subset of parent portfolio allocations with the non-dominated subset of offspring portfolio allocations into a larger set of portfolio allocations;
- (g) passing the larger set of portfolio allocations through a non-crowding filter to identify a reduced subset of portfolio allocations;
- (h) creating a new population of individual portfolio allocations from the reduced subset of portfolio allocations;
- (i) updating the non-dominated portfolio allocations archive with the new population of individual portfolio allocations;
- (j) repeating steps (a) through (i) for a plurality of generations; and
- (k) passing the updated non-dominated portfolio allocations archive through the dominance filter to generate an interim efficient frontier in a portfolio performance space having at least three-dimensions, the interim efficient frontier being used in investment decisions.

28. The method of Claim 27, wherein the non-dominated subset of parent portfolio allocations has a first cardinality.

29. The method of Claim 28, wherein the non-dominated subset of offspring portfolio allocations has a second cardinality that is different than the first cardinality.

30. The method of Claim 29, wherein the larger set of portfolio allocations has a third cardinality that is equal to the first cardinality plus the second cardinality.

31. The method of Claim 30, wherein the reduced subset of portfolio allocations has a fourth cardinality that is less than the third cardinality.

32. The method of Claim 27, wherein steps (a) – (j) are repeated until convergence is achieved or allocated computational cycles are exhausted.

33. The method of Claim 27, wherein the reduced subset of portfolio allocations has a fourth cardinality, and wherein the new population of individual portfolio allocations is exactly equal to the reduced set of portfolio allocations if the fourth cardinality is equal to the initial population of individual portfolio allocations.

34. The method of Claim 27, wherein the reduced subset of portfolio allocations has a fourth cardinality, and wherein the new population of individual portfolio allocations is created by randomly drawing additional individual portfolio allocations from the portfolio allocations archive if the fourth cardinality is less than to the initial population of individual portfolio allocations.

35. The method of Claim 27, wherein the reduced subset of portfolio allocations has a fourth cardinality, and wherein the new population of individual portfolio allocations is created by randomly discarding individual portfolio allocations from the reduced subset of portfolio allocations if the fourth cardinality is greater than to the initial population of individual portfolio allocations.

36. The method of Claim 35, wherein the new population of individual portfolio allocations is created by randomly injecting individual portfolio allocations from the portfolio

allocations archive until the fourth cardinality is equal to a desired number of individual portfolio allocations.

37. A system for multi-objective portfolio analysis using Pareto Sorting Evolutionary Algorithms comprising an efficient frontier processing portion that randomly draws an initial population of individual portfolio allocations that are generated from a portfolio allocations archive by using a combination of linear programming and sequential linear programming algorithms; passes the initial population of portfolio allocations through a dominance filter to identify a non-dominated subset of parent portfolio allocations; commits the non-dominated subset of parent portfolio allocations to a non-dominated portfolio allocations archive; randomly combines matched pairs of parent portfolio allocations to create offspring portfolio allocations; passes the offspring portfolio allocations through the dominance filter to identify a non-dominated subset of offspring portfolio allocations; combines the non-dominated subset of parent portfolio allocations with the non-dominated subset of offspring portfolio allocations into a larger set of portfolio allocations; passes the larger set of portfolio allocations through a non-crowding filter to identify a reduced subset of portfolio allocations; creates a new population of individual portfolio allocations from the reduced subset of portfolio allocations; updates the non-dominated portfolio allocations archive with the new population of individual portfolio allocations; and passes the updated non-dominated portfolio allocations archive through the dominance filter to generate an interim efficient frontier in a portfolio performance space having at least three-dimensions, the interim efficient frontier being used in investment decisions.

38. The system of Claim 37, wherein the non-dominated subset of parent portfolio allocations has a first cardinality.

39. The system of Claim 38, wherein the non-dominated subset of offspring portfolio allocations has a second cardinality that is different than the first cardinality.

40. The system of Claim 39, wherein the larger set of portfolio allocations has a third cardinality that is equal to the first cardinality plus the second cardinality.

41. The system of Claim 40, wherein the reduced subset of portfolio allocations has a fourth cardinality that is less than the third cardinality.

42. The system of Claim 37, wherein the reduced subset of portfolio allocations has a fourth cardinality, and wherein the new population of individual portfolio allocations is exactly equal to the reduced set of portfolio allocations if the fourth cardinality is equal to the initial population of individual portfolio allocations.

43. The system of Claim 37, wherein the reduced subset of portfolio allocations has a fourth cardinality, and wherein the new population of individual portfolio allocations is created by randomly drawing additional individual portfolio allocations from the portfolio allocations archive if the fourth cardinality is less than to the initial population of individual portfolio allocations.

44. The system of Claim 37, wherein the reduced subset of portfolio allocations has a fourth cardinality, and wherein the new population of individual portfolio allocations is created by randomly discarding individual portfolio allocations from the reduced subset of portfolio allocations if the fourth cardinality is greater than to the initial population of individual portfolio allocations.

45. The system of Claim 44, wherein the new population of individual portfolio allocations is created by randomly injecting individual portfolio allocations from the portfolio allocations archive until the fourth cardinality is equal to a desired number of individual portfolio allocations.

46. A computer readable medium for multi-objective portfolio analysis using Pareto Sorting Evolutionary Algorithms comprising an efficient frontier processing portion that randomly draws an initial population of individual portfolio allocations that are generated from a portfolio allocations archive by using a combination of linear programming and sequential linear programming algorithms; passes the initial population of portfolio allocations through a dominance filter to identify a non-dominated subset of parent portfolio allocations; commits the non-dominated subset of parent portfolio allocations to a non-dominated portfolio allocations

archive; randomly combines matched pairs of parent portfolio allocations to create offspring portfolio allocations; passes the offspring portfolio allocations through the dominance filter to identify a non-dominated subset of offspring portfolio allocations; combines the non-dominated subset of parent portfolio allocations with the non-dominated subset of offspring portfolio allocations into a larger set of portfolio allocations; passes the larger set of portfolio allocations through a non-crowding filter to identify a reduced subset of portfolio allocations; creates a new population of individual portfolio allocations from the reduced subset of portfolio allocations; updates the non-dominated portfolio allocations archive with the new population of individual portfolio allocations; and passes the updated non-dominated portfolio allocations archive through the dominance filter to generate an interim efficient frontier in a portfolio performance space having at least three-dimensions, the interim efficient frontier being used in investment decisioning.

47. The computer readable medium of Claim 46, wherein the non-dominated subset of parent portfolio allocations has a first cardinality.

48. The computer readable medium of Claim 47, wherein the non-dominated subset of offspring portfolio allocations has a second cardinality that is different than the first cardinality.

49. The computer readable medium of Claim 48, wherein the larger set of portfolio allocations has a third cardinality that is equal to the first cardinality plus the second cardinality.

50. The computer readable medium of Claim 49, wherein the reduced subset of portfolio allocations has a fourth cardinality that is less than the third cardinality.

51. The computer readable medium of Claim 46, wherein the reduced subset of portfolio allocations has a fourth cardinality, and wherein the new population of individual portfolio allocations is exactly equal to the reduced set of portfolio allocations if the fourth cardinality is equal to the initial population of individual portfolio allocations.

52. The computer readable medium of Claim 46, wherein the reduced subset of portfolio allocations has a fourth cardinality, and wherein the new population of individual portfolio allocations is created by randomly drawing additional individual portfolio allocations from the portfolio allocations archive if the fourth cardinality is less than to the initial population of individual portfolio allocations.

53. The computer readable medium of Claim 46, wherein the reduced subset of portfolio allocations has a fourth cardinality, and wherein the new population of individual portfolio allocations is created by randomly discarding individual portfolio allocations from the reduced subset of portfolio allocations if the fourth cardinality is greater than to the initial population of individual portfolio allocations.

54. The computer readable medium of Claim 53, wherein the new population of individual portfolio allocations is created by randomly injecting individual portfolio allocations from the portfolio allocations archive until the fourth cardinality is equal to a desired number of individual portfolio allocations.

IX. EVIDENCE APPENDIX

No evidence pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132 is/are entered by the Examiner. Accordingly, no evidence is/are relied upon by the Appellant in this paper.

X. RELATED PROCEEDINGS APPENDIX

No related proceedings pursuant to 37 C.F.R. § 41.37(c)(1)(ii) is/are entered by, relied upon, or submitted by the Appellant with this paper.